

## ***E-Book Reader Berbasis Mobile Menggunakan Modifikasi Fitur Face Tracking***

**Saparudin<sup>1</sup>, Reza Firsandaya Malik<sup>2\*</sup>, Rikza Azriyan<sup>3</sup>**

<sup>1,2,3</sup>Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya

Jln Raya Palembang Prabumulih Km.32 Inderalaya, Ogan Ilir, Indonesia

e-mail: <sup>1</sup>saparudin@unsri.ac.id, <sup>2</sup>rezafm@unsri.ac.id, <sup>3</sup>rikzaxtrmsprt@gmail.com

### ***Abstrak***

*Pembaca buku elektronik adalah perangkat lunak untuk membaca buku elektronik pada komputer, dimana navigasi perpindahan halaman biasanya dilakukan dengan menggunakan tombol. Pada makalah ini akan dilakukan pembahasan mengenai navigasi perangkat lunak pembaca buku elektronik berbasis mobile berkamera depan menggunakan teknik Face Tracking yang telah dimodifikasi. Perubahan urutan proses dan pengembangan dari teknik face tracking dengan metode perbandingan warna citra telah dilakukan dalam OpenCV Frameworks. Dari hasil eksperimen didapat bahwa bahwa metode perbandingan warna citra dapat mengeliminasi proses perhitungan yang kompleks pada teknik face tracking dan mampu berjalan dalam memori yang terbatas.*

***Kata kunci:*** Face Tracking, E-book reader, Navigasi, OpenCV Frameworks

### ***Abstract***

*E-book reader is a software for reading electronic books on a computer, where the displacement page navigation is usually done by using the keypad. This paper will discuss the navigation of mobile based e-book reader with front camera by using Modified Face Tracking techniques. The change order process and the development of face tracking techniques with color image comparison method in OpenCV Frameworks have been made. The experimental results obtained that Modified Face Tracking could eliminate the complex calculation process and be able to run in limited memory.*

***Keywords:*** Face Tracking, E-book reader, Navigation, OpenCV Frameworks

### **1. Pendahuluan**

E-book reader adalah perangkat lunak untuk membaca buku elektronik pada media elektronik seperti komputer, smart phone, PDA (Personal Digital Assistant), Tablet PC, dan media – media lain yang mendukung. Sistem navigasi E-book reader selama ini dilakukan dengan memberi aksi pada tombol navigasi, selain itu menggunakan Gesture Recognition yang menerima masukan dari perangkat yang dilengkapi fasilitas touch screen.

Face tracking merupakan jalan alternatif bagi manusia untuk berinteraksi dengan komputer, begitu pula sebaliknya. Dan proses face tracking dapat dilakukan secara waktu nyata, cepat dan efisien tanpa banyak menghabiskan sumber daya sistem [1]. Sebagaimana penelitian yang telah dilakukan oleh Martin Hunke dalam interaksi antara manusia dan komputer, sebuah camcorder diletakkan pada tripod yang dapat dikontrol secara horisontal dan vertikal, diterapkan fitur face tracking [2]. Camcorder tersebut dapat mengikuti wajah seorang pembicara kemanapun ia bergerak dan menyesuaikan konstansitas ukuran wajah pada gambar di camcorder.

Face tracking memungkinkan untuk diterapkan pada perangkat mobile. Menurut penelitian yang telah dilakukan oleh Abdullah Bulbul, untuk mengatasi keterbatasan memori dan CPU power pada perangkat mobile, diperlukan sebuah algoritma yang ringan seperti metode perbandingan warna citra [3]. Dalam makalah ini, akan dibahas mengenai pengembangan sistem navigasi perpindahan halaman E-book reader pada perangkat mobile berkamera depan menggunakan wajah Melalui fitur *face tracking* dengan bantuan OpenCV Frameworks [6].

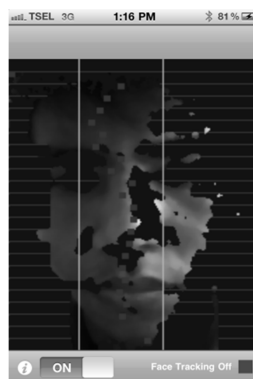
Struktur tulisan ini pada bagian I mengenai pendahuluan yang membahas kegunaan E-book reader dan pengembangan sistem navigasi pada E-book reader, bagian II membahas metodologi, bagian IV membahas hasil dan pembahasan dan bagian V kesimpulan.

## 2. Metodologi Penelitian

### 2.1. Navigasi Perpindahan Halaman E-book Reader

Sistem navigasi perpindahan halaman *e-book reader* pada perangkat mobile dapat beraneka ragam, diantaranya dengan melakukan interaksi *touch* pada *touch screen*, interaksi pada tombol navigasi, dan pemanfaatan *Gesture Recognition*. Pada makalah ini akan dilakukan pembahasan mengenai pengembangan sistem navigasi *e-book reader* berbasis mobile berkamera depan menggunakan fitur *Face Tracking*.

Pada Gambar 1, diasumsikan posisi wajah sebagai inisialisasi awal diletakkan berada di tengah kamera depan. Dengan pergerakan wajah ke arah kanan atau kiri sebagai letak kordinat wajah pada sumbu horinzontal (sumbu x). Dimana kordinat tersebut akan didefinisikan sebagai navigasi perpindahan halaman pada *e-book reader*. Pergerakan wajah ke arah kiri, mengindikasikan navigasi perpindahan ke halaman selanjutnya. Pergerakan wajah ke arah kanan, mengindikasikan perpindahan ke halaman sebelumnya. Seperti yang digambarkan pada Gambar 2 dan Gambar 3.



Gambar 1. Posisi Wajah Berada di Tengah Kamera Depan.



Gambar 2. Pergerakan Wajah ke Arah Kanan

## 2.2. Modifikasi Teknik *Face Tracking*

Metode yang digunakan pada teknik *face tracking* adalah metode perbandingan warna citra, namun dilakukan modifikasi pada urutan proses dan algoritma proses. Metode ini melalui beberapa urutan tahap, yaitu penyesuaian rata-rata cahaya, eliminasi daerah flat, eliminasi daerah yang memiliki fluktuasi, pencarian warna wajah, pencarian hue wajah, eliminasi keluaran *noise*, proses *clipping*, dan proses *rendering*. Perubahan urutan ini dilakukan untuk pengawalan proses deteksi wajah dan diikuti oleh penentuan perkiraan posisi wajah dengan proses *clipping*.



Gambar 3. Pergerakan Wajah ke Arah Kiri

### 2.2.1. Proses Penyesuaian Rata-rata Cahaya

Dalam melakukan proses ini dibutuhkan nilai *Light* dari citra, oleh sebab itu citra harus berada pada *HLS Color Space*. Setelah dilakukan *adjust* pada nilai *Light*, nilai tersebut diletakkan kembali pada citra menggantikan nilai *Light* yang lama.

$$L_p' = \begin{cases} L_p \times \frac{Aim}{Avg}, & L_p < Avg \\ 255 - \left(255 - L_p\right) \times \frac{255 - Aim}{255 - Avg}, & L_p \geq Avg \end{cases} \quad (1)$$

dimana  $Lp'$  adalah nilai *Light* yang baru,  $Lp$  adalah nilai *Light* pada piksel suatu citra,  $Avg$  adalah nilai *Light* rata - rata pada suatu citra, dan  $Aim$  adalah target *Average Light* pada citra dan nilai berskala antara 0 – 255.  $Aim$  ditentukan secara dinamis saat runtime dengan cara :

- a.  $Aim + 40, Avg < (255/4)$
- b.  $Aim - 40, Avg > (255*3/4)$
- c.  $Aim = Avg, Avg \geq (255/4) < (255*3/4)$

### 2.2.2 Proses Eliminasi Daerah Flat

Dalam melakukan proses ini dibutuhkan nilai *Light* dari citra, oleh sebab itu citra harus berada pada *HLS Color Space*. Proses ini melakukan penjumlahan selisih nilai *Light* pada piksel - piksel citra yang terletak berurut dalam jumlah yang ditentukan. Langkahnya sebagai berikut :

$$\sum_{i \in n(i)} |L_i| < th \rightarrow \text{eliminate} \quad (2)$$

dimana  $n(i)$  adalah piksel tetangga dari piksel  $i$ , dan  $th$  adalah nilai *threshold*. Nilai *threshold* ditentukan sebesar 25 untuk 30 piksel yang berurut, dan  $L_i'$  adalah hasil penjumlahan selisih nilai *Light* citra pada citra yang berurutan.

### 2.2.3. Proses Eliminasi Daerah Yang Memiliki Fluktuasi

Untuk melakukan proses ini citra harus berada pada *RGB Color Space*, dan selisih nilai  $R$ ,  $G$ ,  $B$  masing-masing diperhitungkan. Selanjutnya nilai dari selisih masing-masing  $R$ ,  $G$ ,  $B$  tersebut dijumlahkan. Jika jumlahnya lebih besar dari *threshold* dalam jumlah 30 piksel yang berurut, maka bagian ini tereliminasi. *Threshold* ditentukan sebesar 750. Langkahnya sebagai berikut :

$$\sum_{i \in n(i)} |R_i| + |G_i| + |B_i| > th \rightarrow \text{eliminate} \quad (3)$$

dimana  $n(i)$  adalah piksel tetangga dari  $i$  dan  $th$  adalah nilai *threshold*.  $|R_i|$ ,  $|G_i|$ , dan  $|B_i|$  adalah nilai hasil dari selisih piksel  $i$  terhadap piksel  $n(i)$ .

### 2.2.4. Proses Pencarian Warna Wajah

Untuk melakukan proses ini citra harus berada pada  $R, G, B$  *Color Space*. Dalam pencarian warna kulit, perbandingan rasio warna merah terhadap biru haruslah berada pada rentang 1:1 hingga 3:1. Berdasarkan pada penafsiran tersebut, warna diluar rasio tersebut dieliminasi karena tidak dapat dikatakan sebagai warna kulit. Akan tetapi proses ini akan berjalan tidak optimal jika latar belakang berwarna kayu, furniture, dan coklat.

### 2.2.5. Proses Pencarian Hue Wajah

Dalam melakukan proses ini citra harus berada pada  $H, L, S$  *Color Space*. Dalam proses ini dilakukan pembagian nilai  $H$  *Color Space* ke dalam beberapa *cluster*. Dengan begitu, pada masing-masing *cluster*, dilakukan pencarian jumlah maksimum yang

diberikan nilai *H Color Space* yang paling sering digunakan dan diasumsikan sebagai cluster nilai *Hue* pada wajah. Selanjutnya dilakukan eliminasi terhadap bagian - bagian lain yang tidak termasuk ke dalam *cluster* tersebut. Jumlah *cluster* ditentukan sebanyak 40 *cluster*. Langkahnya sebagai berikut :

- a. Diasumsikan posisi wajah pada citra diatas  $\frac{1}{2}$  atau minimal  $\frac{1}{3}$  dari bagian citra.
- b. Dilakukan pencarian hue terbanyak yg digunakan, hue terbanyak yg digunakan diasumsikan sebagai hue wajah
- c. Membagi Hue menjadi 40 cluster dan menghitung jumlah total pixel yang nilai Hue nya berada pada cluster tersebut.
- d. Jumlah cluster terbanyak diasumsikan sebagai nilai hue pada wajah. Kemudian dilakukan perluasan cluster dengan mengambil 5 cluster di depannya, dan 5 cluster di belakangnya sebagai hue wajah.

### 2.2.6. Proses Eliminasi Keluaran Noise, Proses Clipping, dan Proses Rendering

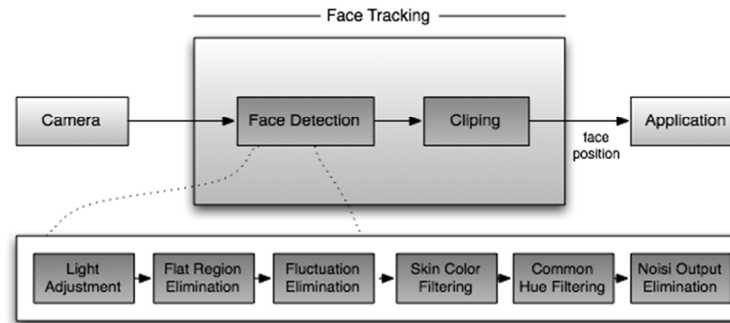
Setelah seluruh proses eliminasi dilakukan, proses eliminasi terakhir adalah mengeliminasi *noisy region* pada citra dengan menggunakan *median filter*. Proses selanjutnya adalah *Clipping*, yaitu melakukan perhitungan letak posisi wajah pada garis *scan* yang telah ditentukan. Pada makalah ini garis *scan* yang akan digunakan adalah membuat garis clipping sepanjang sumbu y. Semakin banyak garis clipping, maka semakin akurat penentuan kecendrungan posisi wajah. Kemudian membagi citra menjadi 3 daerah yaitu: kiri, tengah, dan kanan. Membagi flag navigasi menjadi 4 yaitu: flag Kiri, flag kanan, flag tengah, dan flag error. Dimana masing – masing Flag menjelaskan posisi wajah:

- a. Jika berada titik tengah wajah berada di kiri, increment flag kiri.
- b. Jika berada titik tengah wajah berada di kanan, increment flag kanan.
- c. Jika berada di tengah increment flag tengah.
- d. Jika berada pada titik  $x=0$  increment flag error.
- e. Terakhir, melakukan perhitungan kecendrungan posisi wajah dengan membandingkan jumlah flag (antara flag kiri, kanan, dan tengah). Flag error tidak ikut dihitung karena dianggap sebagai titik error (tidak ada wajah).

Proses Rendering *eBook* dilakukan setelah proses clipping, yaitu saat setelah wajah melakukan perpindahan, dan kembali ke regiont tengah. Proses ini dilakukan agar proses penangkapan wajah tidak dilakukan berulang – ulang dalam mengikuti proses penangkapan frame.

### 3. Hasil dan Pembahasan

Tiap - tiap proses dibawah ini dilakukan secara berurutan, seperti yang ditunjukkan pada Gambar 4.



Gambar 4. Diagram Urutan Proses Modifikasi Teknik Face Tracking

### 3.1. Proses Penyesuaian Rata-rata Cahaya

Proses penyesuaian rata-rata cahaya dilakukan untuk membantu penyesuaian pencahayaan pada citra di berbagai lingkungan sesuai dengan persamaan (1). Hasil dari proses penyesuaian rata-rata cahaya ditunjukkan seperti pada Gambar 5. Langkahnya sebagai berikut :

- Ubah citra ke dalam bentuk *HLS Color Space*.
- Dapatkan *Avg* pada citra
- Dapatkan *Lp* yang terletak di kiri atas citra
- Jika  $Lp < Avg$  maka  $Lp' = Lp * (Aim/Avg)$
- Jika  $Lp \geq Avg$  maka  $Lp' = (255 - (255 - Lp) * ((255 - Aim)/(255 - Avg)))$
- Letakkan nilai  $Lp'$  menggantikan nilai  $Lp$  yang lama.



Gambar 5. Penyesuaian Rata-rata Cahaya

### 3.2 Proses Eliminasi Daerah Flat

Pada proses ini dilakukan eliminasi pada citra yang memiliki nilai Light tidak berubah sepanjang jumlah urutan piksel dikarenakan lekukan dari wajah manusia sesuai dengan persamaan (2). Hasil dari proses eliminasi daerah flat ditunjukkan pada Gambar 6. Langkahnya sebagai berikut :

- Ubah citra ke dalam bentuk *HLS Color Space*.

- b. Tentukan selisih nilai *Light* antara piksel  $i$  dengan pikse  $n(i)$ , begitu pula piksel  $n(i)$  dengan pikse tetangganya hingga 30 piksel yang berurutan.
- c. Tentukan nilai  $Li'$  pada 30 piksel yang berurutan
- d. Jika  $Li' < th$  maka eliminasi piksel tersebut.



Gambar 6. Eliminasi Daerah Flat

### 3.2. Proses Eliminasi Daerah Yang Memiliki Fluktuasi

Pada proses ini dilakukan eliminasi pada piksel yang memiliki warna yang berulang sepanjang urutan jumlah piksel. Bagian ini tidak dapat dikatakan sebagai warna wajah karena warna wajah selalu berubah sepanjang arah tertentu menurut persamaan (3). Hasil dari proses ini ditunjukkan pada Gambar 7. Untuk melakukan proses eliminasi daerah yang memiliki fluktuasi, langkah-langkah yang harus dilakukan adalah sebagai berikut:

- a. Ubah citra ke dalam bentuk *RGB Color Space*
- b. Tentukan  $|Ri''|$ ,  $|G''|$ , dan  $|Bi''|$ .
- c. Jumlahkan  $|Ri''|$ ,  $|G''|$ , dan  $|Bi''|$ .
- d. Jika  $|Ri''| + |G''| + |Bi''| > th$  maka eliminasi 30 piksel yang berurut tersebut.



Gambar 7. Eliminasi Daerah Yang Memiliki Fluktuasi

### 3.3. Proses Pencarian Warna Wajah

Pada Proses ini dilakukan eliminasi pada piksel yang tidak termasuk warna kulit. Berdasarkan penelitian sebelumnya [3], [4], dan [5], warna biru pada kulit bervariasi dalam rentang yang begitu luas. Oleh sebab itu pada warna biru tidak efektif pada seluruh warna dibandingkan warna merah. Sesuai dengan penelitian [3], [4], dan [5], perbandingan warna merah-hijau pada warna kulit berubah antara 1:1 hingga 3:1. Berdasarkan penafsiran tersebut, bagian selain warna tersebut harus dieliminasi karena bukan merupakan warna kulit. Seperti yang ditunjukkan Gambar 8.



Gambar 8. Elminasi Selain Warna Kulit

### 3.4. Proses Pencarian Hue Wajah

Pada proses ini diasumsikan porsi wajah pada citra diatas  $\frac{1}{2}$  atau minimal  $\frac{1}{3}$  dari citra. Berdasarkan asumsi ini, dilakukan pembagian nilai *Hue* kedalam 40 *cluster*. Seperti yang ditunjukkan pada Gambar 9. Langkah-langkah yang dilakukan adalah:

- Lakukan pembagian 40 *cluster*, yaitu;  
Cluster 1 : 0 - 8  
Cluster 2 : 9 - 17  
Cluster 3 : 18 - 26  
Hingga cluster 40 : 351 - 360
- Lakukan iterasi nilai *H Color Space* pada seluruh piksel, dan tetapkan masing - masing piksel pada rentang *cluster* yang sesuai
- Lakukan iterasi pencarian *cluster* terbanyak yang digunakan pada 40 *cluster* dalam penentuan rentang nilai *Hue* terbanyak yang digunakan.
- Eliminasi seluruh piksel yang tidak termasuk dalam rentang *cluster* tersebut.



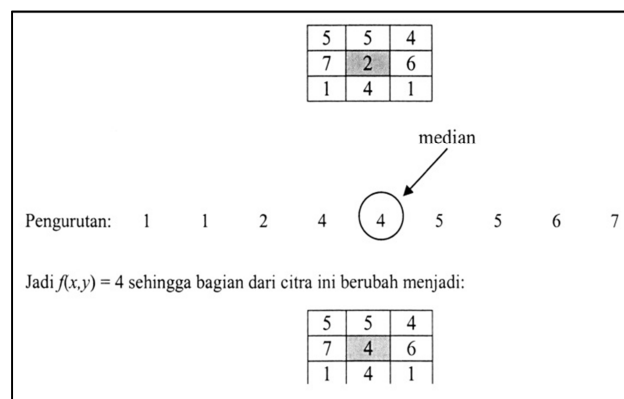
Gambar 9. Proses Pencarian Hue



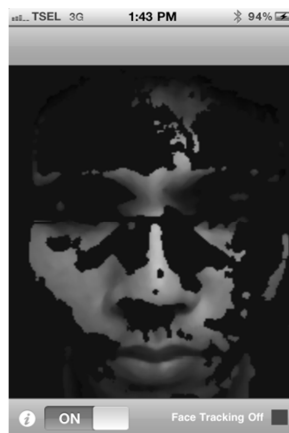
### 3.5. Proses Eliminasi Keluaran Noise

Proses ini merupakan proses eliminasi piksel yang terkahir, karena masih ada bagian kecil piksel yang bukan warna kulit yang tidak ikut tereliminasi. Diaplikasikan *median filter* untuk mengeliminasi bagian piksel ini. Langkah-langkahnya seperti digambarkan pada Gambar 10, sedangkan hasil dari proses *median filter* ditunjukkan pada Gambar 11., yaitu:

- Urutkan nilai piksel tetangga dari piksel yang akan dilakukan median filter.
- Cari nilai yang terletak di tengah dari piksel yang telah diurutkan.
- Letakkan nilai piksel median pada piksel yang dilakukan median filter.



Gambar 10. Proses Median Filter



Gambar 11. Hasil *Median Filter*

### 3.6. Proses Clipping

Dalam proses ini dilakukanlah seluruh perhitungan perkiraan kordinat wajah pada sumbu x. kordinat wajah terdeteksi dengan *scanning* sepanjang *clip* garis horizontal yang telah ditentukan. Semakin banyak *clip* garis yang diberikan, semakin akurat pula perkiraan kordinat wajah. Seperti yang ditunjukkan pada Gambar 12, dilakukan *scanning* sepanjang garis - garis horizontal yang diberikan untuk mendapatkan titik perkiraan posisi wajah. Setiap titik memiliki 3 wilayah kordinat, yaitu kordinat kiri, tengah, dan kanan. Dilakukan perhitungan jumlah titik yang menempati

tiap - tiap wilayah kordinat. Jumlah titik terbanyak pada suatu wilayah kordinat menjelaskan navigasi perpindahan halaman yang akan dilakukan.

Untuk mengoptimalkan kebutuhan memori, proses *rendering* perpindahan halaman eBook dilakukan hanya disaat titik - titik kordinat wajah terbanyak kembali berada pada wilayah kordinat tengah.



Gambar 12. Proses Clipping

#### 4. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah, modifikasi pengurutan proses dan algoritma proses pada teknik *face tracking* menggunakan metode perbandingan warna citra dengan bantuan OpenCV Frameworks sehingga dapat mendeteksi pergerakan wajah dan menghindari proses perhitungan yang kompleks pada perangkat mobile yang memiliki keterbatasan memori. Metode ini masih tetap bergantung pada kondisi latar belakang lingkungan sekitar yang tertangkap pada kamera depan perangkat mobile.

#### Penghargaan

Terimakasih kepada Abdullah Bulbul, Departement of Computer Engineering, Bilkent University, Ankara, Turkey dan Fakultas Ilmu Komputer, Universitas Sriwijaya, Palembang, Indonesia

#### Referensi

- [1] G. R. Bradski. 1998. Computer Vision Face Tracking for Use in a Perceptual User Interface. *Intel Technology Journal*, 1998.
- [2] M. Hunke dan A. Waibel. 1994. *Face Locating and Tracking for Human-Computer Interaction*. Proceeding of the 28th Asilomar Conf. on Signals, System dan Computers, pages 1277-1281, 1994.
- [3] A. Bulbul. 2009. *A Face Tracking Algorithm for User Interaction in Mobile Devices*. Proceeding of International Conference on CyberWorlds: IEEE, 2009.
- [4] J. Brdan dan J. Mason. *A Comparative Assessment of Three Approaches to Pixel-Level Human Skin Detection*. Proceeding of IEEE International Conference Pattern Recognition, vol. 1, pp. 1056-1059, Sept. 2000.
- [5] V. Vezhnevets, V. Sazanov, dan A. Danereeva. 2003. *A Survey on Pixel-Based Skin Color Detection Techniques*. Proceeding of GRAPHICON03, pp. 85-92, 2003

- 
- [6] G. R Bradski, dan Andrian Keahler. Learning OpenCV Computer Vision With The OpenCV Library, O-Reilly, 2008.